



AMENDMENTS TO THE CLAIMS

The following listing of claims replaces all previous listings.

Listing of Claims:

1. (Previously presented) A system for automated generation of one or more query language statements comprising:

a syntax pattern selector module for selecting, in an automated process, a syntax pattern corresponding to a desired function provided to the syntax pattern selector module and a syntax standard for use in generating the one or more query language statements;

a statement assembly module for populating the syntax pattern in an automated process with an argument data set associated with a desired data set provided to the statement assembly module as part of the process of generating the one or more query language statements; and

whereby at least one query language statement is assembled to be run against a data source to return the desired data set.

2. (Original) The system of claim 1, wherein the syntax pattern selector module selects the syntax pattern from a plurality of syntax patterns corresponding to a plurality of database management systems.

3. (Original) The system of claim 1, wherein the syntax pattern selector module selects the syntax pattern from a plurality of syntax patterns based upon at least one selection variable.

4. (Previously presented) A system for generating one or more query language statements comprising:

a syntax pattern selector module for selecting a syntax pattern corresponding to a desired

function and a syntax standard;

a statement assembly module for populating the syntax pattern with an argument data set associated with a desired data set;

a structure generator module for generating a query structure based on the desired data set, the query structure providing a basis for identifying the desired function to be used by the syntax pattern selector module; and

whereby at least one query language statement is assembled to be run against a data source to return the desired data set.

5. (Previously presented) A system for generating one or more query language statements comprising:

a syntax pattern selector module for selecting a syntax pattern corresponding to a desired function and a syntax standard;

a statement assembly module for populating the syntax pattern with an argument data set associated with a desired data set;

a function identifier module for identifying a functional element corresponding to the desired function and at least one syntax pattern; and

whereby at least one query language statement is assembled to be run against a data source to return the desired data set.

6. (Previously presented) A system for generating one or more query language statements comprising:

a syntax pattern selector module for selecting a syntax pattern corresponding to a desired function and a syntax standard;

a statement assembly module for populating the syntax pattern with an argument data set

associated with a desired data set;

an argument generator module for identifying the argument data set associated with the desired data set; and

whereby at least one query language statement is assembled to be run against a data source to return the desired data set.

7. (Original) The system of claim 6, wherein the argument generator module identifies the argument data set based upon a syntax description associated with the desired function.

8. (Original) The system of claim 1, wherein the system is a component in an online analytical processing system, a reporting system, a business intelligence system, or a data mining system.

9. (Original) The system of claim 1, further comprising a plurality of driver modules, each of the driver modules including at least one syntax pattern associated with a selected database management system.

10. (Currently amended) A computer-implemented method of generating one or more query language statements to be run against a one or more data sources, comprising the steps, performed by a computer system, of:

selecting a syntax pattern corresponding to a desired function provided as an input and a syntax standard for use in generating the one or more query language statements;

populating the syntax pattern with an argument data set associated with a desired data set provided as an input identifying the data set on which to operate from the data source as part of generating the one or more query language statements; and

wherein, the populated syntax pattern is used to generate one or more query language

statements ~~which may be run~~ runnable against one or more data sources to return the desired data result set.

11. (Original) The method of claim 10, wherein the step of selecting the syntax pattern includes selecting the syntax pattern from a plurality of syntax patterns corresponding to a plurality of database management systems.

12. (Original) The method of claim 10, wherein the step of selecting the syntax pattern includes selecting the syntax pattern from a plurality of syntax patterns based upon at least one selection variable.

13. (Original) The method of claim 10, further comprising the step of generating a query structure based on the desired data set, the query structure providing a basis for identifying the desired function to be used in selecting the syntax pattern.

14. (Currently amended) A method of generating one or more query language statements to be run against a one or more data sources, comprising the steps of:
selecting a syntax pattern corresponding to a desired function and a syntax standard;
populating the syntax pattern with an argument data set associated with a desired data set from the data source;

identifying a functional element corresponding to the desired function and at least one syntax pattern, the functional element providing a basis for selecting the syntax pattern; and

wherein, the populated syntax pattern comprises one or more query language statements ~~which may be run~~ runnable against one or more data sources to return the desired data result set.

15. (Original) The method of claim 10, further comprising the step of identifying at least one selection variable for selecting the syntax pattern from a plurality of syntax patterns.

16. (Currently amended) A method of generating one or more query language statements to be run against a one or more data sources, comprising the steps of:

- selecting a syntax pattern corresponding to a desired function and a syntax standard;
- populating the syntax pattern with an argument data set associated with a desired data set from the data source;
- identifying the argument data set associated with the desired data set; and

wherein, the populated syntax pattern comprises one or more query language statements ~~which may be run~~ runnable against one or more data sources to return the desired data result set.

17. (Original) The method of claim 16, wherein the step of identifying the argument data set includes identifying the argument data set based upon a syntax description associated with the desired function.

18. (Original) The method of claim 10, wherein the method is executed in an online analytical processing systems, a reporting system, a business intelligence system, or a data mining system.

19. (Original) The method of claim 10, wherein the step of selecting the syntax pattern includes accessing a plurality of driver modules including at least one syntax pattern, each of the plurality of driver modules corresponding to a selected database management system.

20. (Currently amended) A tangible medium having a ~~processor~~ computer readable program code embodied therein for generating one or more query language statements through an automated computer-implemented method comprising:

- code for causing the processor to identify a functional element corresponding to a desired function for use in generating the one or more query language statements;
- code for causing the processor to identify an argument data set associated with a desired

data set and the identified functional element as part of generating the one or more query language statements;

code for causing the processor to select a syntax pattern corresponding to the functional element; and

code for causing the processor to populate the selected syntax pattern with the identified argument data set to assemble at least one query language statement to be run against a data source.

21. (Previously presented) A method of generating a query language statement comprising the steps of:

defining a syntax pattern accessible to a system for generating a query language statement;

accessing the defined syntax pattern from the system to generate a query language statement; and

wherein the system does not need to be recompiled as a result of defining the syntax pattern.

22. (Original) The method of claim 21, wherein the syntax pattern is associated with a selected database management system.

23. (Currently amended) A tangible medium having a ~~processor~~ computer readable program code embodied therein for generating one or more query language statements comprising:

code for causing the processor to identify a functional element corresponding to a desired function;

code for causing the processor to identify an argument data set associated with a desired

data set and the identified functional element;

code for causing the processor to select a syntax pattern corresponding to the functional element;

code for identifying a functional element corresponding to the desired function and at least one syntax pattern, the functional element providing a basis for selecting the syntax pattern; and

code for causing the processor to populate the selected syntax pattern with the identified argument data set to assemble at least one query language statement to be run against a data source.

24. (Currently amended) A tangible medium having a ~~processor~~ computer readable program code embodied therein for generating one or more query language statements comprising:

code for causing the processor to identify a functional element corresponding to a desired function;

code for causing the processor to identify an argument data set associated with a desired data set and the identified functional element;

code for causing the processor to select a syntax pattern corresponding to the functional element;

code for identifying the argument data set associated with the desired data set; and

code for causing the processor to populate the selected syntax pattern with the identified argument data set to assemble at least one query language statement to be run against a data source.